Enabling the 5-Level Paging Feature of Microsoft Windows Server 2022 on Lenovo ThinkSystem Servers

Planning / Implementation

Windows operating systems use address-translation support called paging. Paging translates virtual address (aka. Linear address) used by the OS, into physical address, which is used to access memory (or memory mapped I/O).

The Page table is a data structure that the memory manager creates and maintains, and the CPU translates virtual address into physical address. Each page of virtual address space is associated with a system-space structure called a page table entry (PTE), which contains the physical address to which the virtual one is mapped.

4-Level Paging

Address translation on x64 architecture is similar to x86, but with a fourth level added which limits virtual address to 48 bits. The components that make up this 48-bit virtual address and the connection between the components for translation purposes are shown in Figure 1.

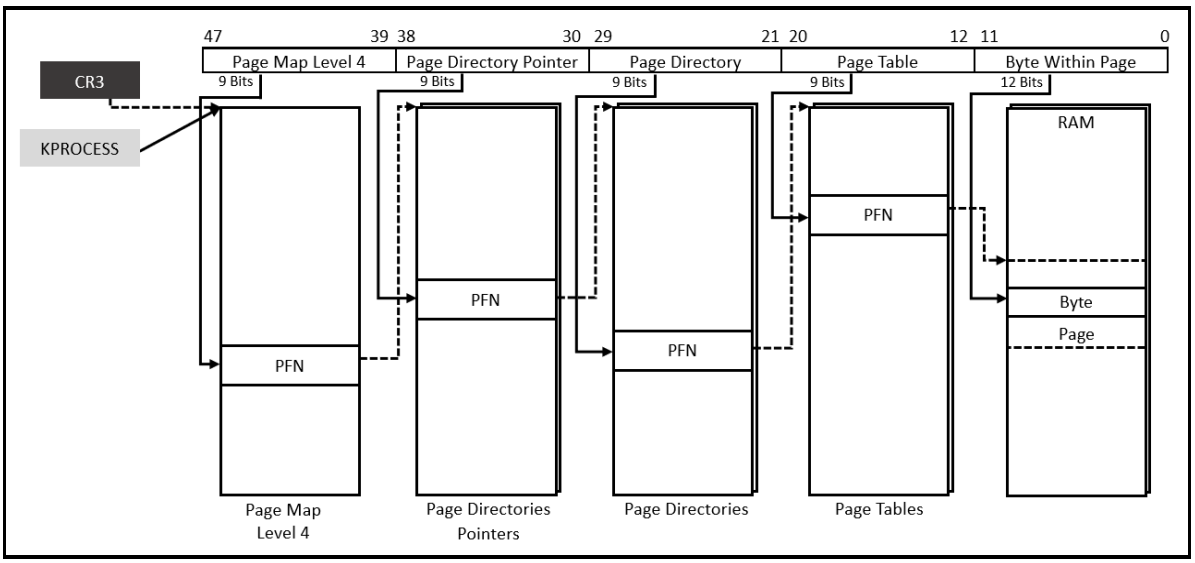

Figure 1. x64 address translation – 4-Level paging

EXPERIMENT: Viewing the base address of the page table entry (PTE) on 4-level paging by WinDBG

The `nt!MiGetPTEaddress` routine is part of the Windows NT kernel's Memory Manager, and it is used to retrieve the address of the PTE corresponding to a given virtual address.

```
8: kd> uf nt!MiGetPTEaddress
nt!MiGetPteAddress:
fffff801`2febb634 48c1e909        shr     rcx,9
fffff801`2febb638 48b8f8ffffff7f000000 mov rax,7FFFFFFF8h
fffff801`2febb642 4823c8          and     rcx,rax
fffff801`2febb645 48b80000000000eaffff mov rax,0FFFFEA0000000000h
fffff801`2febb64f 4803c1          add     rax,rcx
fffff801`2febb652 c3              ret
```

Then, using the kernel debugger !address command to identify virtual memory region of the PTE.

```
8: kd> !address FFFFEA0000000000
Usage:
Base Address:           ffffea00`00000000
End Address:            ffffea80`00000000
Region Size:            00000080`00000000
VA Type:                PageTables
  Hex:     ffffea00`00000000
  Binary:  11111111 11111111 11101010 00000000 00000000 00000000 00000000 00000
000
   Bit 63 - No execute
Bit 48~62 - Reserved
Bit 47~12 - PFN
Bit 11 - Write (software)
Bit 10 - prototype (software)
Bit 9 - Copy on write (software)
Bit 8 - Global
Bit 7 - Large page
Bit 6 - Dirty
Bit 5 - Accessed
Bit 4 - Cache disabled
Bit 3 - Write through
Bit 2 - Owner
Bit 1 - Write
Bit 0 - Valid
```

5-Level Paging

However, with the increasing need for larger address spaces to accommodate complex applications and data structures, a more sophisticated approach became necessary. Introducing 5-level paging, a groundbreaking memory management mechanism introduced in Windows Server 2022.

5-level paging is an extension that alleviates the limitation of 57 linear address bits (as depicted in Figure 2). This architectural advancement enables Windows Server 2022 to effectively manage massive virtual address spaces, supporting up to 128 petabytes of virtual memory.

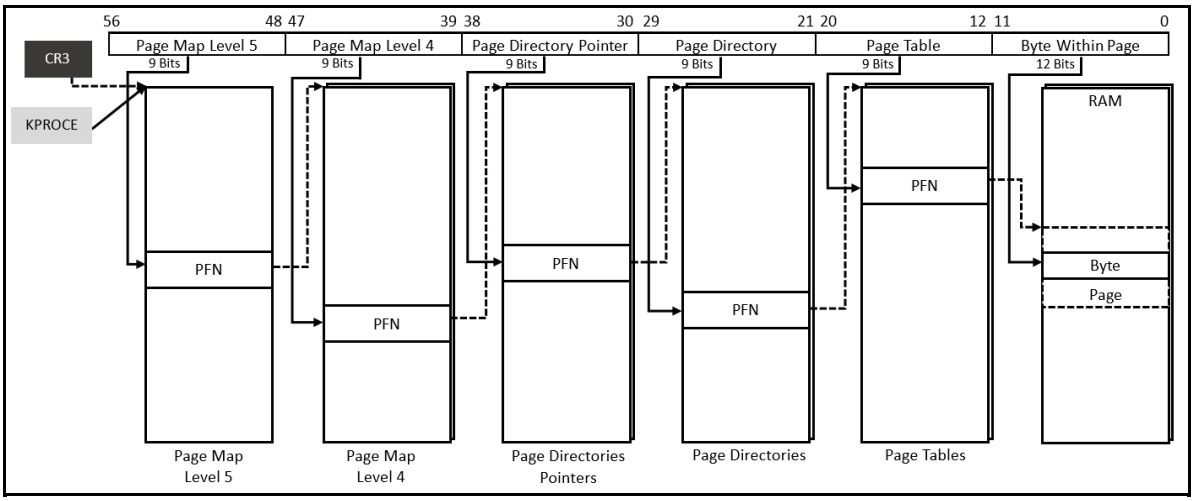

Figure 2. x64 address translation – 5-Level paging

The 5-level paging is designed to enhance system scalability, enabling Windows Server 2022 to meet the escalating memory requirements of contemporary applications, including large databases, virtualized environments, and high-performance computing workloads.

EXPERIMENT: Viewing the base address of the page table entry (PTE) on 5-level paging by WinDBG.

```
0: kd> uf nt!MiGetPTEaddress
nt!MiGetPteAddress:
fffff803`4a0261c0 48c1e909        shr     rcx,9
fffff803`4a0261c4 48b8f8ffffffffff0000 mov rax,0FFFFFFFFFFF8h
fffff803`4a0261ce 4823c8          and     rcx,rax
fffff803`4a0261d1 48b80000000000002dff mov rax,0FF2D000000000000h
fffff803`4a0261db 4803c1          add     rax,rcx
fffff803`4a0261de c3              ret
0: kd> !address FF2D000000000000
Usage:
Base Address:           ff2d0000`00000000
End Address:            ff2d0080`00000000
Region Size:            00000080`00000000
VA Type:                PageTables
  Hex:     ff2d0000`00000000
  Binary:  11111111 00101101 00000000 00000000 00000000 00000000 00000000 00000
000
Bit 63 - No execute
Bit 57~62 - Reserved
Bit 56~12 - PFN (extend from bit 48 to 56 due to PML5)
Bit 11 - Write (software)
Bit 10 - prototype (software)
Bit 9 - Copy on write (software)
Bit 8 - Global
Bit 7 - Large page
Bit 6 - Dirty
Bit 5 - Accessed
Bit 4 - Cache disabled
Bit 3 - Write through
Bit 2 - Owner
Bit 1 - Write
Bit 0 - Valid
```

We can see the virtual memory address region of the PTE is up to 57 bits virtual address if the 5-level paging is enabled.

Supported Lenovo servers

To support 5-level paging, servers must have the functionality enabled in UEFI. The following servers all support 5-level paging with Windows Server 2022:

- ThinkSystem V2 with 3rd Gen Intel Xeon Scalable processors
  - ThinkSystem SD630 V2
  - ThinkSystem SD650 V2
  - ThinkSystem SR630 V2
  - ThinkSystem SR650 V2
- ThinkSystem V3 with 4th or 5th Gen Intel Xeon Scalable processors
  - ThinkSystem SD550 V3
  - ThinkSystem SD530 V3
  - ThinkSystem SR630 V3
  - ThinkSystem SR650 V3
  - ThinkSystem ST650 V3

Note: Lenovo ThinkSystem servers with AMD EPYC processors are currently not enabled to support 5-level paging even though the processors support it.

Enabling 5-level Paging in UEFI

To support 5-level paging you will need to set the following in System Setup.

1. Boot the server to System Setup by pressing F1 when prompted
2. Navigate to **System Settings** → **Processors**
3. Change **Limit CPU PA to 46 bits** to **Disabled** as shown in Figure 3.

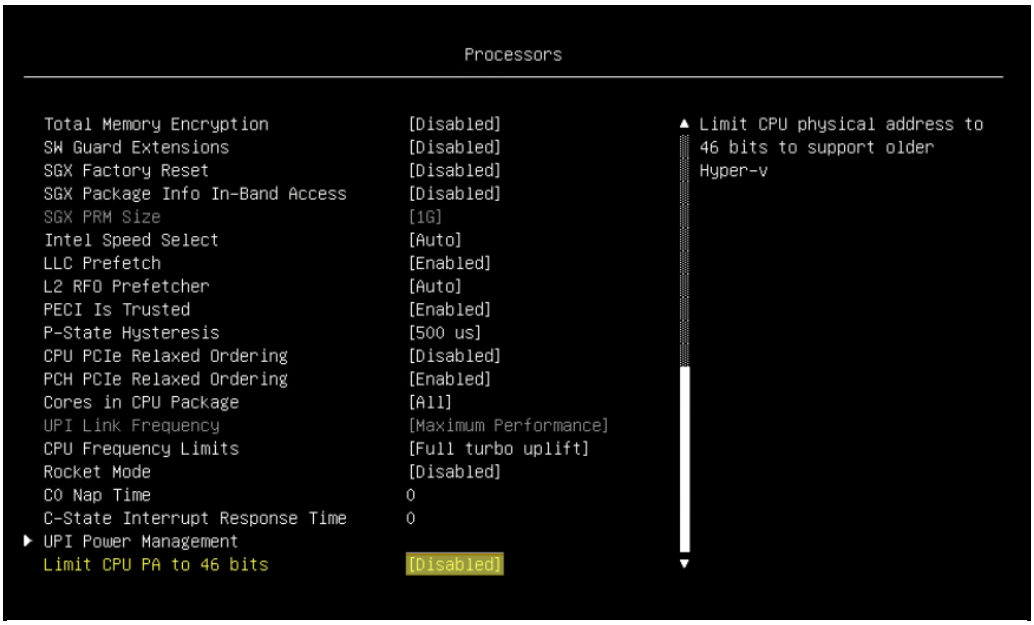

Figure 3. Set Limit CPU PA to 46 bits to Disabled

Enabling 5-level Paging in Windows Server 2022

After the UEFI setting is ready for 5-level paging support, you will need to enable a setting in the operating system.

To enable 5-level paging in the BCD store use the following steps:

1. Open a Command Prompt windows as an Administrator, Figure 4.

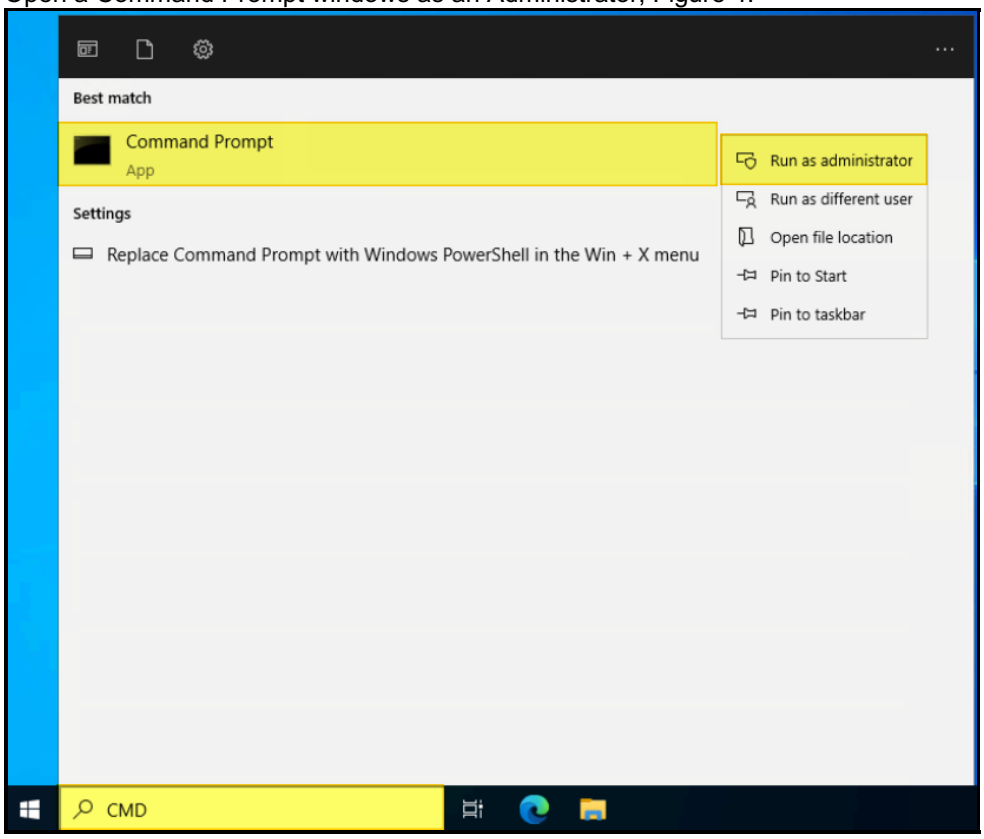

Figure 4. Run Command Prompt as administrator

2. Type the command **bcdedit /set linearaddress57 optin**, Figure 5.

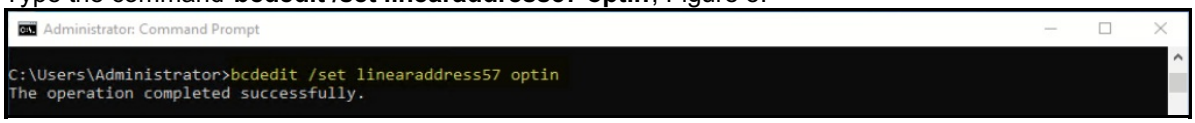

Figure 5. Enable 5-level paging by BCDEdit

3. Reboot the system to make the changes take effect.

Checking the status of 5-level paging

To confirm 5-level paging feature is properly configured and running, do the following steps:

1. To confirm LA57 (5-level paging) is enabled, issue `bcdedit` in a Command Prompt and confirm that linearaddress57 is `optin` as shown in Figure 6.

```
C:\Users\Administrator>bcdedit

Windows Boot Manager
--------------------
identifier              {bootmgr}
device                  partition=\Device\HarddiskVolume1
path                    \EFI\Microsoft\Boot\bootmgfw.efi
description             Windows Boot Manager
locale                  en-US
inherit                 {globalsettings}
bootshutdowndisabled    Yes
default                 {current}
resumeobject            {c8be1540-5801-11ee-8247-e5bf35f37eb1}
displayorder            {current}
toolsdisplayorder       {memdiag}
timeout                 30

Windows Boot Loader
-------------------
identifier              {current}
device                  partition=C:
path                    \Windows\system32\winload.efi
description             Windows Server
locale                  en-US
inherit                 {bootloadersettings}
recoverysequence        {c8be1542-5801-11ee-8247-e5bf35f37eb1}
displaymessageoverride  Recovery
linearaddress57         optin
recoveryenabled         Yes
isolatedcontext         Yes
allowedinmemorysettings 0x15000075
osdevice                partition=C:
```

Figure 6. BCD store check

2. Processors that support 5-level paging allow software to set an enabling bit, CR4.LA57[bit 12]. If CR4.LA57 = 1, 5-level paging is used. Therefore, to confirm the enabling bit, you will need to download a utility like the RW tool for the confirmation. Download RW tool from: http://rweverything.com/downloads/RwPortableX64V1.7.zip
3. Launch the application Rw.exe, Figure 7.

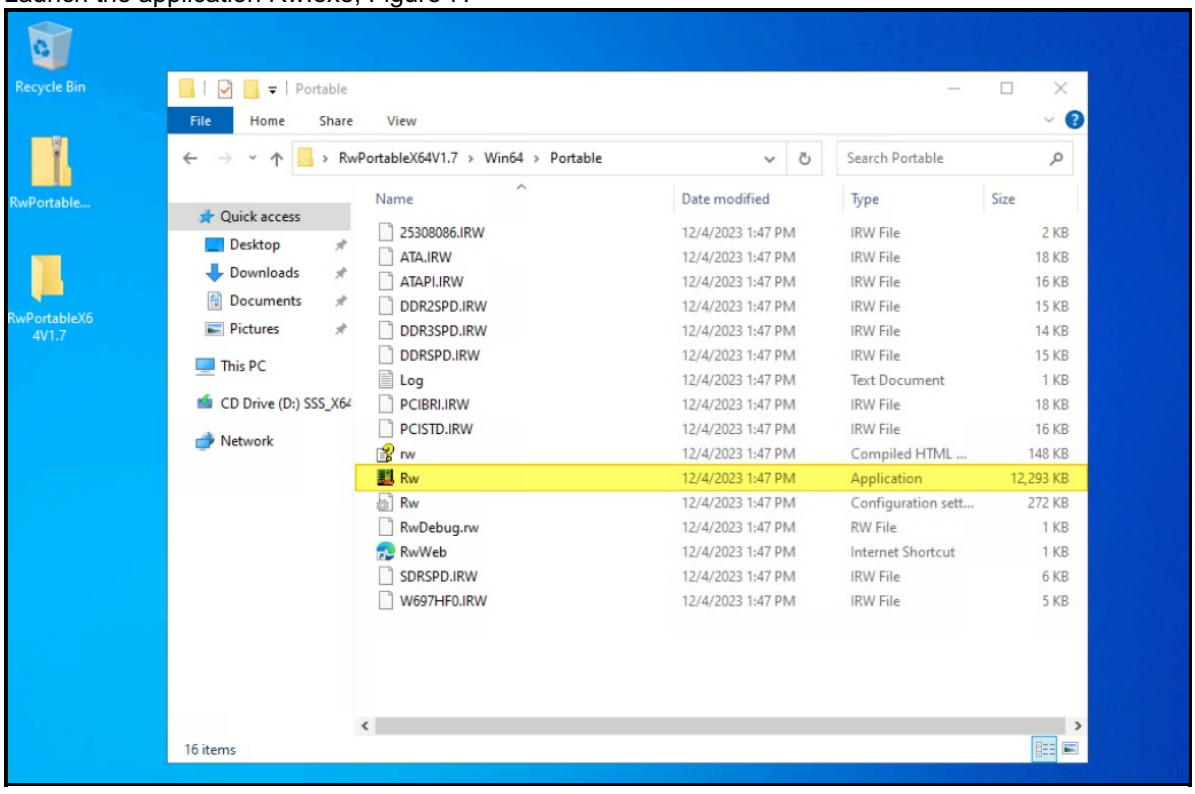

Figure 7. RW tool package

4. Click **Command** as shown in Figure 8.

Figure 8. RW tool - Command

5. Issue `rdcr 4` in the Command (as shown in Figure 9), then check if the bit 12 is 1.

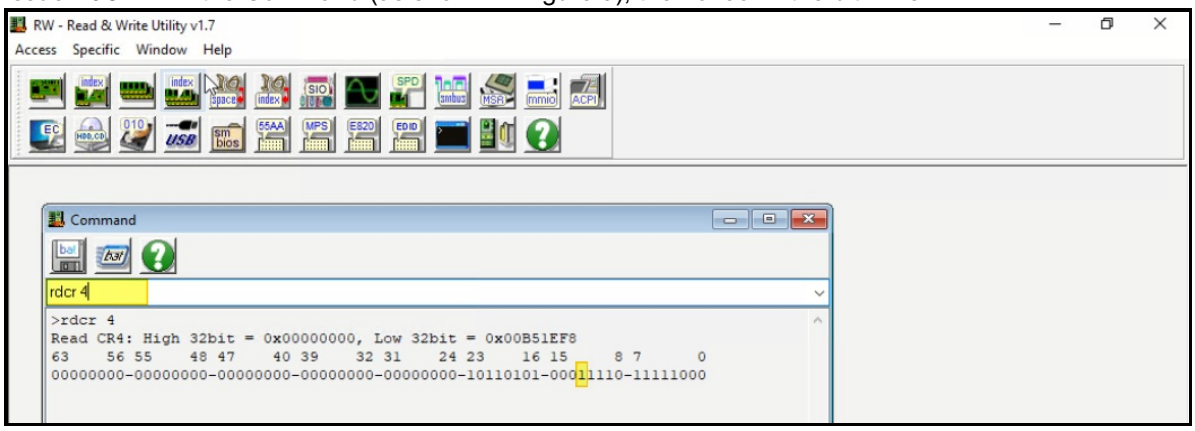

Figure 9. Check CR4.LA57 [bit 12] via RW tool

References

For more information, see these resources:

- Pavel, Y., Alex I., Mark E. R., & David A. S. (2017). Windows Internals Part1: System architecture, processes, threads, memory managemenet, and more, 7th Washington, United States of America: Microsoft Press.
- Intel White Paper. (2017). 5-Level Paging and 5-Level EPT. Retrieved from https://www.intel.com/content/www/us/en/content-details/671442/5-level-paging-and-5-level-ept-white-paper.html (August 9, 2023)
- (2023). AMD64 Architecture Programmer's Manual Volume 2: System Programming. Retrieved from https://www.amd.com/content/dam/amd/en/documents/processor-tech-docs/programmer-references/24593.pdf (January 28, 2024)

Author

Wewe Chang is a Windows Engineer for the Lenovo Infrastructure Group, based in Taipei, Taiwan. She has more than 9 years of experience with Windows kernel and user mode debugging.

Special thanks to the following people for their contributions and suggestions:

- Micahel Miller, Advisory Engineer, ThinkAgile Development
- Boyong Li, Senior Engineer, OS Enablement
- Gary Cudak, OS Architect, ThinkAgile Development
- David Watts, Lenovo Press

Related product families

Product families related to this document are the following:

- Microsoft Windows
- Processors

Notices

Lenovo may not offer the products, services, or features discussed in this document in all countries. Consult your local Lenovo representative for information on the products and services currently available in your area. Any reference to a Lenovo product, program, or service is not intended to state or imply that only that Lenovo product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any Lenovo intellectual property right may be used instead. However, it is the user's responsibility to evaluate and verify the operation of any other product, program, or service. Lenovo may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not give you any license to these patents. You can send license inquiries, in writing, to:

Lenovo (United States), Inc.
8001 Development Drive
Morrisville, NC 27560
U.S.A.
Attention: Lenovo Director of Licensing

LENOVO PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some jurisdictions do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. Lenovo may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice.

The products described in this document are not intended for use in implantation or other life support applications where malfunction may result in injury or death to persons. The information contained in this document does not affect or change Lenovo product specifications or warranties. Nothing in this document shall operate as an express or implied license or indemnity under the intellectual property rights of Lenovo or third parties. All information contained in this document was obtained in specific environments and is presented as an illustration. The result obtained in other operating environments may vary. Lenovo may use or distribute any of the information you supply in any way it believes appropriate without incurring any obligation to you.

Any references in this publication to non-Lenovo Web sites are provided for convenience only and do not in any manner serve as an endorsement of those Web sites. The materials at those Web sites are not part of the materials for this Lenovo product, and use of those Web sites is at your own risk. Any performance data contained herein was determined in a controlled environment. Therefore, the result obtained in other operating environments may vary significantly. Some measurements may have been made on development-level systems and there is no guarantee that these measurements will be the same on generally available systems. Furthermore, some measurements may have been estimated through extrapolation. Actual results may vary. Users of this document should verify the applicable data for their specific environment.

This document, LP1911, was created or updated on March 5, 2024.

Send us your comments in one of the following ways:

- Use the online Contact us review form found at:
  https://lenovopress.lenovo.com/LP1911
- Send your comments in an e-mail to:
  comments@lenovopress.com

This document is available online at https://lenovopress.lenovo.com/LP1911.

Trademarks

Lenovo and the Lenovo logo are trademarks or registered trademarks of Lenovo in the United States, other countries, or both. A current list of Lenovo trademarks is available on the Web at https://www.lenovo.com/us/en/legal/copytrade/.

The following terms are trademarks of Lenovo in the United States, other countries, or both:
Lenovo®
ThinkAgile®
ThinkSystem®

The following terms are trademarks of other companies:

Intel® and Xeon® are trademarks of Intel Corporation or its subsidiaries.

Linux® is the trademark of Linus Torvalds in the U.S. and other countries.

Microsoft®, Microsoft Press®, Windows Server®, and Windows® are trademarks of Microsoft Corporation in the United States, other countries, or both.

Other company, product, or service names may be trademarks or service marks of others.